Restricted Turing Machines

Martin Čermák, Jiří Koutný and Alexander Meduna

Deparment of Information Systems
Faculty of Informatin Technology
Brno University of Technology, Faculty of Information Technology
Božetěchova 2, Brno 612 00, Czech Republic

BRNO UNIVERSITY OF TECHNOLOGY

FIT

FACULTY OF INFORMATION TECHNOLOGY

Advanced Topics of Theoretical Computer Science

FRVŠ MŠMT FR2581/2010/G1

Part I

Restricted Turing Machines

Restricted Turing machines are

- easier to deal with when compared to their general versions,
- as powerful as the general versions.

Types of restriction

1. restrictions placed upon computation,
2. restrictions placed upon the size.

Computational Restrictions

We require these machines to work deterministically:

- From any configuration, they can make no more than one move.

Definition

Turing Machine is *deterministic* if it represents a deterministic rewriting system.

Turing Machines are equivalent if they define the same language.

Definition

Turing Machines are *equivalent* if their languages coincide.

Theorem

For every Turing Machine M, there exists an equivalent deterministic Turing Machine D.

Definition

Let M be a Turing Machine. If from $\chi \in_M X$, M can make no move, then χ is a *halting configuration* of M.

Theorem

For every deterministic Turing Machine D, there exists an equivalent deterministic Turing Machine $M = (\Sigma, R)$ such that ${}_M Q$ contains two new states, $\blacklozenge$ and $\blacksquare$, which do not occur on the left-hand side of any rule in ${}_M R$, ${}_M F = \{\blacksquare\}$, and

1. every halting configuration $\chi \in_M X$ has the form $\chi = \triangleright q u \triangleleft$ with $q \in \{\blacklozenge, \blacksquare\}$, and every non-halting configuration $v \in_M X$ satisfies $\{\blacklozenge, \blacksquare\} \cap aplh(v) = \emptyset$;
2. on every input string $x \in_M \triangle^*$, M performs one of these three kinds of computation:
   - $\triangleright \blacktriangleright x \triangleleft \Rightarrow^* \triangleright \blacksquare u \triangleleft$, where $u \in_M \Gamma^*$,
   - $\triangleright \blacktriangleright x \triangleleft \Rightarrow^* \triangleright \blacklozenge v \triangleleft$, where $v \in_M \Gamma^*$,
   - M never enters any halting configuration.

Convention

- Turing Machine M has the properties of previous Theorem.
- Let ${}_{TM}\Psi$ be the set of all these machines.
- Let ${}_{TM}\Phi = \{L(M) | M \in {}_{TM}\Psi\}$ be the *family of Turing languages*.

Convention

Let $M \in {}_{TM}\Psi$ and $x \in {}_{M}\triangle^*$. We say that

- *M accepts x* iff on x, M makes a coputation on the form (i),
- *M rejects x* iff on x, M makes a coputation on the form (ii),
- *M halts x* iff it accepts or rejects x,
- *M loops* on *x* iff it performs a computation of the form (iii),
- State $\blacksquare$ is *accepting* state and $\triangleright\blacksquare u\triangleleft$ is *accepting* configuration,
- State $\blacklozenge$ is *rejecting* state and $\triangleright\blacklozenge u\triangleleft$ is *rejecting* configuration.

Every $L \in_{TM}\Phi$ is accepted by $O \in_{TM}\Psi$ that never rejects any input x.

- either O accepts x or O loops on x.

We cannot reformulate this result so O never accepts any input.

- because the language of any Turing Machine that accepts no input is empty.

We cannot reformulate this result so O never loops.

- because ${}_{TM}\Phi$ contains languages accepted only by Turing Machines that loop on some inputs.

Theorem

Let $M \in_{TM}\Psi$. Then, there is $O \in_{TM} \Psi$ such that $L(M) = L(O)$ and O never rejects any input.

This result has crucial consequences in computer science as a whole.

Place a limit on the number of tape symbols in Turing Machines.

Theorem

Let $D \in_{TM} \Psi$ width $card({}_{D}\triangle) \geq 2$. Then, there is $M \in_{TM} \Psi$ with ${}_{M}\Gamma = {}_{D}\triangle \cup \{\square\}$.

Corollary

Let $D \in_{TM} \Psi$. Then, there exists $M \in_{TM} \Psi$ with ${}_{M}\Gamma = \{a, b, \square\} \cup {}_{D}\triangle$.

We can also place a limit on the number of states in Turing Machines.

Theorem

Let $D \in_{TM} \Psi$. Then, there exists $M \in_{TM} \Psi$ with $card({}_{M}Q) \leq 3$.

By simultaneously placing a limit on both the number of non-input tape symbols and the number of states, we decrease the power of Turing Machines.

References

- Wayne Goddard.
  Introducing the Theory of Computation.
  Jones Bartlett Publishers, 2008.
- Jeffrey D. Ullman John E. Hopcroft, Rajeev Motwani.
  Introduction to Automata Theory, Languages, and Computation.
  Addison Wesley, 2006.
- Dexter C. Kozen.
  Automata and Computability.
  Springer, 2007.
- Dexter C. Kozen.
  Theory of Computation.
  Springer, 2010.
- John C. Martin.
  Introduction to Languages and the Theory of Computation.
  McGraw-Hill Science/Engineering/Math, 2002.

Thank you for your attention!

End